

**REMARKS**

**I SPECIFICATION**

A. The disclosure stands objected to because of an embedded hyperlink. The hyperlink has been removed.

B. Drawing corrections were requested due to informalities. Replacement drawing sheets for Figures 1-7 have been submitted.

**II CLAIMS**

Claims 1-32 are pending. Claims 1, 13, and 28 have been amended.

**A. 112 Rejection**

Claim 6 stands rejected under 35 U.S.C. § 112, second paragraph for insufficient antecedent basis due to the limitation “one or more tables” recited in the claim. Applicants respectfully disagree. “One” and “one or more” are indefinite articles unlike the definite articles “said” or “the.” Applicants respectfully request that the rejection be withdrawn.

**B. 101 Rejection**

Claims 1-19 stand rejected under 35 U.S.C. 101 as being directed to non-statutory subject matter. Claim 1 has been amended to recite “computer implemented” as suggested by the Examiner. Claims 2-19 depend on claim 1 and as such are amended as well.

**C. 102 Rejections**

Claims 1-15, 18-19, 22, 24-29 and 31-32 stand rejected under 35 U.S.C. 102 as being anticipated by U.S. Patent 6,708,173 to Behr et al. (hereafter Behr).

1. Amended claim 1 recites “[a] computer implemented process for materializing a trace in a markup language syntax, the process comprising: creating a meta-language grammar; creating

a trace grammar in which the trace grammar complies with rules of the meta-language grammar; generating one or more traces compliant with the trace grammar; parsing the one or more traces; identifying interrelationships within the one or more traces; and generating a new version of the one or more traces using a markup language syntax.” In contrast, Behr discloses a tracing program that provides debugging capabilities for multiple components over the internet. For example, Behr specifically discloses:

The present invention overcomes the disadvantages of the prior art by providing a method of and apparatus for debug [sic.] of multiple component applications for access of a legacy data base management system via [sic.] Internet terminal.

(Behr, column 4, lines 64-67) underline added.

More specifically, claim 1 recites “creating a meta-language grammar.” However, Behr merely discusses transferring transaction data over the internet. The cited passage of Behr discloses:

the present invention must first provide an interface herein referred to generically as a gateway, which translates transaction data transferred from the user over the internet in HTML format into a format from which data base management system commands and inputs may be generated.

(Behr, column 5, lines 1-5) underline added.

This passage of Behr discusses making a transaction database system web-enabled to allow transaction data to be transferred over the internet. A web-enabled system does not read on “*creating a meta-language grammar*” as recited in claim 1.

Claim 1 further recites “creating a trace grammar in which the grammar complies with rules of the meta-language grammar.” In contrast, Behr discloses writing traces to a common file. More specifically, the cited passage of Behr discloses:

provides a web based interface that allows a developer to create a web based service that joins tables... This service renders the resulting table to the web...  
...the preferred mode of the present invention employs a tracing facility called “Utrace”, which allows components from one or more applications to write trace information to a common trace file.

(Behr, column 6, lines 27-38). underline added.

This passage of Behr merely shows the trace data from different components being transferred over the internet and saved in the same file. However, writing trace information from multiple components to a common file is not the claimed limitation. Behr does not disclose “*creating a trace grammar* in which the grammar complies with rules of the meta-language grammar.”

For at least the above reasons, Behr does not disclose “[a] computer implemented process for materializing a trace in a markup language syntax, the process comprising: creating a meta-language grammar; creating a trace grammar in which the trace grammar complies with rules of the meta-language grammar; generating one or more traces compliant with the trace grammar; parsing the one or more traces; identifying interrelationships within the one or more traces; and generating a new version of the one or more traces using a markup language syntax“ as recited in claim 1.

2. Claims 32 recites substantially similar limitations as claim 1, and as such, is patentable over Behr for at least the same reasons.

3. Claims 2-15, and 18-19 depend on claim 1 and as such are patentable over Behr for at least the same reasons.

**D. 103 Rejections**

1. Claim 7 stands rejected under 35 U.S.C. 103 as being unpatentable over Behr in view of U.S. Patent 6,754,890 to Berry et al. (hereafter Berry).

Claim 7 recites “[t]he process of claim 6 in which one or more tables comprises hash tables corresponding to keywords in the one or more traces.” Claim 7 depends on claim 6, which in turn depends in claim 1. Behr does not disclose the limitations in claim 1. Berry does not cure these deficiencies.

Berry provides a tracing mechanism where the process identifier of a process within a program being monitored is included in the trace file. For example, Berry specifically discloses:

A method of monitoring execution performance of a program is provided. A process identifier associated with a process within a program is determined, and a trace output file is created for the process such that the file name of the trace output file contains the process identifier. Trace records are generated in response to events within the process. The trace records associated with the process are then written to the trace output file associated with the process. Multiple processes may then be associated with unique trace output files simultaneously. Using this methodology, multiple instances of JVMs may be executing simultaneously, and each JVM may be generating trace records through a profiler. However, the origin of the trace records, as identified by the process identifier of the JVM, is used to place the trace information into a file that is identified through the use of the same process identifier.

(Berry, column 2, lines 63- column 3 ,line 11) underline added.

Berry further discloses:

With reference now to FIG. 4, a block diagram depicts components used to profile processes in a data processing system. A trace program 400 is used to profile processes 402. Trace program 400 may be used to record data upon the execution of a hook, which is a specialized piece of code at a specific location in a routine or program in which other routines may be connected. Trace hooks are typically inserted for the purpose of debugging, performance analysis, or enhancing functionality. These trace hooks are employed to send trace data to trace program 400, which stores the trace data in buffer 404. The trace data in buffer 404 may be stored in a file for post-processing. . . .

With reference now to FIG. 5, a diagram depicts various phases in profiling the processes active in an operating system. . . .

An initialization phase 500 is used to capture the state of the client machine at the time tracing is initiated. This trace initialization data includes trace records that identify all existing threads, all loaded classes, and all methods for the loaded classes. Records from trace data captured from hooks are written to indicate thread switches, interrupts, and loading and unloading of classes and jitted methods. Any class which is loaded has trace records that indicate the name of the class and its methods. In the depicted example, four byte IDs are used as identifiers for threads, classes, and methods. These IDs are associated with names output in the records. A record is written to indicate when all of the start up information has been written.

Next, during the profiling phase 502, trace records are written to a trace buffer or file. Trace records may originate from two types of profiling actions-- event-based profiling and sample-based profiling. In the present invention, the trace file may have a combination of event-based records, such as those that may

originate from a trace hook executed in response to a particular type of event, e.g., a method entry or method exit, and sample-based records, such as those that may originate from a stack walking function executed in response to a timer interrupt, e.g., a stack unwind record, also called a call stack record.  
(Berry, column 9, line 19 to column 10, line 2) underline added.

These passages of Berry teach a tracing method that can store identification information about the source so that the originating process can be identified and state information about the client. However, Berry specifically lacks any information regarding “keywords in the one or more traces” as recited by claim 7.

Further, Berry does not disclose “creating a meta-language grammar” or “creating a trace grammar” as recited in claim 7 through its dependence on claims 6 and 1, and as such, does not cure the deficiency of Behr.

Behr and Berry, neither alone nor together, teach, disclose or suggest the claim limitations of claim 1. As such, Behr and Berry cannot be used to preclude patentability of claim 7 under 35 U.S.C. § 103.

2. Claims 16-17, 20, 30, and 31 stand rejected under 35 U.S.C. 103 as unpatentable over Behr in view of U.S. Patent 6,654,749 to Nashed (hereafter Nashed).

a. Claim 20 recites “[a] system for materializing a trace having markup language syntax, comprising: a first mechanism that receives one or more trace grammars, the one or more trace grammars modifiable within rules of a meta-language grammar; a parser to parse one or more traces complying with the one or more trace grammars; a second mechanism to build one or more semantic networks based upon interrelationships for the one or more traces; and a manifestation mechanism to generate a new version of the traces to include a hyperlink based upon the one or more semantic networks.” Behr does not disclose these limitations as demonstrated in section C of this paper. Nashed does not cure these deficiencies. Nashed discloses a search engine. For example, Nashed specifically discloses:

In accordance with the present invention, method and system for electronically searching information databases of information sources, which can be accessed for free or on a subscription fee basis, provide for access to information on a topic of interest using a search engine which searches information databases whose data records have been indexed into index fields, such as title, full text content and classification category with a plurality of selections, and where indexing data is stored at an indexed database coupled to the search engine.

(Nashed, column 2, lines 40-49) underline added.

Nasjed further teaches:

In one preferred embodiment, the server engine includes a query server containing a search processor which performs searching of the indexed database based on the search query entered and expansion words generated from the search query using semantic network expansion.

(Nashed, column 3, lines 17-21) underline added.

These passages of Nash detail a search engine that searches indexed database records using semantic network expression. Nashed does not disclose “a first mechanism that receives one or more trace grammars, the one or more trace grammars modifiable within rules of a meta-language grammar; a parser to parse one or more traces complying with the one or more trace grammars; a second mechanism to build one or more semantic networks based upon interrelationships for the one or more traces; and a manifestation mechanism to generate a new version of the traces to include a hyperlink based upon the one or more semantic networks” as recited in claim 20 , and as such, does not cure the deficiencies of Behr.

As such, Behr and Nashed, neither alone nor together, teach ,disclose, or suggest the limitations in claim 20 and cannot be used to preclude patentability of these claims under 35 U.S.C. § 103.

b. Claims 22-31 depend on claim 20, and as such, are patentable over Behr and Nashed for at least the same reasons.

c. Claims 16-17 depend on claim 1, and as such, are patentable over Behr and Nashed for at least the same reason.

CONCLUSION

On the basis of the above remarks, reconsideration and allowance of the claims is believed to be warranted and such action is respectfully requested. If the Examiner has any questions or comments, the Examiner is respectfully requested to contact the undersigned at the number listed below.

The Commissioner is authorized to charge any fees due in connection with the filing of this document to Bingham McCutchen's Deposit Account No. 50-2518, referencing billing number **7017812001**. The Commissioner is authorized to credit any overpayment or to charge any underpayment to Bingham McCutchen's Deposit Account No. 50-2518, referencing billing number **7017812001**.

Respectfully submitted,  
Bingham McCutchen LLP

Dated: May 19, 2005

By: Janet D. Chance  
Janet D. Chance  
Reg. No. 55,048

Three Embarcadero Center, Suite 1800  
San Francisco, CA 94111-4067  
Telephone: (650) 849-4904  
Telefax: (650) 849-4800